

Location Privacy: A Comparison of Technologies and Approaches*

Michael J. May
Department of Computer and Information Science
University of Pennsylvania
mjmay@seas.upenn.edu

Abstract

Geolocation aware systems are maturing, however, technologies for ensuring user privacy have not kept up pace. Many approaches have been put forward to ensure user location privacy, each with its own strengths and weaknesses. This paper discusses the issue of geolocation privacy in general and analyzes four systems that address the problem with respect to three location privacy scenarios.

1 Introduction

“You are here.”

The most important piece of information on a map is the little dot indicating where the viewer is. Without that point of reference, the rest of the map is fairly useless. Up until recently maps were all manual devices - users figure out their location and then discover information about their locale through examination of the map. Technology has now reached the point where it has turned the tables - telling the users where they are and feeding them information based on that. To distinguish map information from other forms of location information - virtual, online, network, etc - information about the geographic location of a device or person is termed *geolocation information*.

As geolocation technologies mature, it becomes important to identify how users can be offered both privacy and convenience. In approaching the question of location privacy a few existential questions arise. The first is defining the problem: What does it mean to have a breach of location privacy? How does it differ from a normal breach of information privacy? The

second is coming up with a consistent approach to addressing the issue. Are privacy violations different for each particular implementation, protocol, or usage of geolocation services? Are there common themes and solutions? The third is designing the trust model for geolocation privacy systems. What kinds of threats are being addressed? Where should the user’s trust lie? In a server? In client software? With the observer? Or perhaps nowhere at all?

Having defined the important questions, some answers must be found so that a good idea of the requirements can be decided upon. With that infrastructure, it is easier to be better judges about what technologies are helpful toward the goals and what are orthogonal. Through this investigation ideas about what a location privacy system ought to look like come out. The experience of investigating existing systems helps future designers avoid pitfalls that others have already solved.

2 Scope

First, a definition for location privacy is required. Modern ideas of invisibility are practically oriented. People buy tinted glass for their cars. Celebrities wear dark sunglasses as they walk on the street. Those methods aren’t completely secure - the car has a license plate and the sunglass wearer could be identified by other personal details. Still, both methods afford a level of privacy to the casual observer, preventing one kind of privacy violation. The advantage of geolocation services and systems is that they augment existing standards for discovering the location of objects and provide uniform methods for leveraging that information. Therefore, in this paper, the scope of geolocation privacy is restricted to management of data that is obtained in a non-public manner. In light of that, a breach of privacy is an attacker gaining access to some private location information

*This work was prepared in partial fulfillment of the University of Pennsylvania’s Department of Computer and Information Science Written Preliminary Exam. It was presented and approved in November 2003 by committee members Carl A. Gunter, Zachary Ives (chair), and Stuart Stubblebine.

that would not have been accessible otherwise[6].

With that definition, it is easy to discuss how location privacy breaches differ from standard privacy breaches. Two location privacy goals may be defined: (1) preventing someone from knowing where a person was and (2) preventing someone from knowing the current location of the person. The first type of privacy requires prevention of outsiders from being able to reveal old location data. The second, however, only requires securing current information. Data persistence is a question that must be addressed elsewhere, but this distinction has a say in determining how secure location data must be. For example, if the concern is only with real time data, then quick, less secure methods of encryption may be used. However, if long term data privacy is needed, more secure methods must be used. Note, however, that the concern is with location privacy, not user invisibility. Securing total invisibility is beyond the scope of this paper because even the presence of a user in an encrypted location table reveals existence. Scope is restricted to anonymity and data privacy.

3 Related problems

With that problem scope, different types and implementations of location tracking may be discussed to give a perspective on location privacy. Four cases of virtual or on-line location and location history information leveraging are examined to help better discuss the kind of geolocation solutions that can be designed.

Bluejacking Many new cellular phones are able to communicate with each other using the Bluetooth short-distance messaging protocol. Users have discovered ways of sending anonymous messages using Bluetooth to nearby phones as either jokes or spam. The practice is called *Bluejacking*[3] and is receiving reasonable media attention [9].

Cookie-based tracking Cookies [13], small text strings that are passed to and from internet browsers by web sites, are a method of mimicking state in HTTP environment that is fundamentally stateless. When a browser loads a web page, the server may pass along a short, sometimes encrypted text *cookie* that will be stored by the browser and sent back with each subsequent HTTP request. Web advertisers have learned to manage stored cookies to serve as user identifiers, tracking where the cookie gets sent

and serving web advertisements that fit users' browsing habits

Click stream tracking Along a slightly different tack than the cookie based method, web servers can send out advertising based on click stream [4] history – the path of pages and links chosen in order to reach a particular web page. One level of click stream history can be discovered by looking at the http-referrer field sent along with each request most browsers send to web servers. Multiple levels can be stored in hidden fields in web page content.

Web Beacons Another method of web location tracking is the use of web beacons. Web beacons, also called Web bugs, pixel tags, or clear GIFs, are single pixel graphics that are stored on an advertiser's server and linked to from participating web sites [4]. When a browser sends a web request to download a web beacon, it may pass along connection information, http-referrer information, time of day, and cookie information to the advertiser's server.

These and other forms of *virtual location* based advertising have been around for a while. With the move toward geolocation based systems and location data interoperability and availability, advertisers will begin sending advertisements based on the physical location of a user. It is not hard to imagine a situation where a user device's location is identified and it is immediately sent numerous advertisements for nearby services and shops.

Three important lessons can be brought out from the above location leveraging systems.

First, Bluejacking relies upon a mistake in the implementation of a few Bluetooth enabled phones. By following a method described on the web, it is possible to scan for nearby Bluetooth enabled phones and send them messages anonymously. There is little to stop rogue users from designing their own devices to send Bluetooth spam, but it was not the intention of the phone manufacturers to allow that capability on their phones. When designing a geolocation based system, it is essential that no piece of it can be hijacked by rogue users.

Second, all four methods of tracking mentioned above can be stopped by elite users, those who know how to restrict their systems in an advanced way. Normal users, however, are not privy to such knowledge and are therefore not able to go about their business with privacy. They are more likely to leave a complex system in its default configuration than

to experiment with different settings to make things more private. As such, a system must be designed that is both simple to use and easy to configure. Ideally a system that is private by default should be made, leaving it up to the user to open up secured settings as desired.

Third, it is clear from the above cases that users trust companies and web sites until they hear otherwise. If users knew that a particular web site tracked their personal information and did malicious things with the data that they collect, they would hesitate to visit that web site. In reality, though, normal users visit sites at will, only considering privacy practices when certain types of information are requested from them. In designing a geolocation privacy system, it is advantageous to leverage that inherit trust users have in technology and make sure that they don't have reason to lose that trust.

4 Scenarios

Three geolocation based services scenarios are now described which are used below to discuss the various advantages and disadvantages that arise in four different location privacy management systems. Each scenario has different requirements, so a geolocation privacy system that addresses the issues for each scenario is desired. Through this exercise it will become clearer what the requirements for a real location privacy system are.

Scenario A: National Parks user tracking In order to better allocate money for roads and rangers at different parks, the National Parks system benefits from knowing the travel habits of its visitors. Conventional methods of tracking include counting entries and exits and putting counter strips on roadways, but in some cases finer grained data is desired. The Parks system can track the real time travel patterns of visitors by embedding RFID tags, small chips that send out radio messages when queried by collector devices, in National Parks passes or by tracking the data transmitted by users as they tote their cell phones and wireless devices around. The Parks system is concerned with general usage patterns in the park, what hours certain trails are popular, what roads are being over used, what ranger stations are being understaffed, etc. They want to accurately track user patterns, but don't care about particular visitor data. Visitors might be upset by the notion

of being tracked, but will go along with it if they are assured that their private data will not be divulged.

Scenario B: Rental Car Company tracking

Rental cars often come equipped with Global Positioning System (GPS) receivers that offer services such as on demand maps and directions. In addition to real time tracking, GPS logs can be used to track the places that the car was taken. Suppose a car rental company wants to retrieve location information and do real time tracking for three purposes: (a) car inventory tracking, (b) enforcing rules on the car rental agreement (e.g. no driving across state lines), and (c) discovering usage patterns for particular vehicles (e.g. How much driving time was spent on highways and how much on streets).

In all of the cases the driver is being tracked secondarily to the car itself, so it is desirable to separate the driver from the vehicle data when doing analysis. In the first case the goal is to not track the location of the car at all while it is rented out. Privacy of the car's location must be ensured even when it is parked, barring a solution that just reveals the car's location when it's stopped. In the middle case, the aim is to track the car only to a very coarse granularity, in-state versus out-of-state. A system must be created that gives the flexibility to assure the driver that only the coarse data is being tracked and retained, not more specific location or customer identification data.

Scenario C: Location Services for Wireless Devices

With the increase of 802.11 compliant network services (WiFi connections), it is possible to gain rough location information without additional hardware. Several different proposals have been made, but all focus on the same idea: discovering location from WiFi base stations. Options include roughly correlating the nearest base station with physical location, triangulation of base station signals, and assisted GPS lookups. In any case, that location information can be used to create interesting services. Travellers can receive maps and lists of nearby restaurants, gift shops, and other services. Visitors to office complexes or institutions can receive schedules of their appointments and directions for how to get to them. The information can be personalized or generic depending on what kind of system is desired. Users will want a system that is easy to turn off and control so they will not get undesired messages. They also want assurance that their loca-

tion information will not be abused or misused by the service providers.

The above scenarios are a reasonable set of services that companies might like to deliver. Seeing how each technique meets or fails to meet the above scenario requirements will give a good feeling for the usefulness, strengths, and weaknesses of each technique.

5 Four techniques for location privacy preservation

Four different approaches that can be used to create location privacy systems are now described. First each approach is discussed in depth and then compared and contrasted to the different approaches with respect to the requirements defined in the above scenarios. First is the geolocation privacy model put forward by the Internet Engineering Task Force (IETF)[1] working group called Geopriv [2]. Second is the geolocation privacy language described by Einar Snekkenes [11]. Third is a data mining concept put forth by Rakesh Agrawal and Ramakrishnan Srikant called Privacy Preserving Data Mining [5] that can be used to ensure location privacy. Last is an XML[12] database security mechanism put forth by Gerome Miklau and Dan Suciu called Cryptographically Enforced Conditional Access XML [8]. The last two methods are privacy preserving mechanisms that help achieve the desired geolocation privacy requirements.

5.1 Geopriv

The Geopriv system is XML based and focuses on access rules and a trusted location server. Its goal is to allow people to let others track their location through location (data) objects that they publish while maintaining some user controls. Users define rules both on the location server and embedded in the location object that restrict how the data can be redistributed and retained and how accurate the information released to specific recipients is. Geopriv's goal is a set of languages and protocols that allow users to publish their location information on particular servers, have those servers securely distribute location information to authorized individuals, and maintain control over how others use the geolocation data.

Here are some important terms that the Geopriv working group uses:

- Location Generator: The device that creates location information and publishes it.

- Location Object: The object-oriented data structure that contains raw location information along with some rules.
- Location Server: A server that collects location objects from many different location generators and distributes them based on rules defined both in the object and in external rule sets.
- Rule Maker: The person or entity that decides what sort of filtering and retention rules to associate with the location objects provided from a particular location generator. Often the rule maker is the owner of the location generator.
- Rule Holder: A server that holds the rules that have been provided by the rule maker. The rules are kept securely so that unauthorized outsiders can not view them. The rule holder acts as a persistent, always available site for location servers to query and get rules for managing the location objects of particular location generators.
- Location Recipient: The person or entity that queries the location server to get location information about a particular location generator.
- Sighting: An event wherein the location generator sends out a location object that is received by the location server. For example, a device may send out a location object every 5 minutes to indicate its location. Each time the location server receives that object, it has made a sighting of the device.

The location object is filled out by the location generator and contains the basics for passing along location information. It also may be partially filled out by a location recipient and sent to a location server as a query. The system is still in its requirements phase. So far the working group has put forward a list of what data must be included in the location object. According to the requirements, the following fields must be included:

- Target identifier: Unique ID for the location generator or device being tracked.
- Location recipient identity: ID for the person or entity receiving the location information.
- Location recipient credential: Method that the location recipient(s) are using to identify themselves. Some form of authentication must be specified in this field, though the exact type is still to be determined.

- Location recipient proof of possession of the credential
- Location field: The geographic coordinates, civil location information, etc., that is used to identify the physical location of the device. Optionally, the object's motion and direction may be included.
- Location data type: The type of the location data in the above field. Some default form of location information must be supported by all versions and implementations of the Geopriv system. Whether that format is latitude/longitude, civil location, or something else is still a matter of debate and development. Some form of *delta location*, a message that indicates just a relative change in location as opposed to absolute location, must be permitted for efficiency and security.
- Timing information: When the sighting took place and how long the information contained in the above fields is valid.
- Version: The version of the location object. This field is important to allow for future extensions to the location object definition.
- Redistribute: A single flag included with the location object to indicate permission or prohibition to redistribute it.

The working group plans that some default rules will be kept and passed along with the object while a more complete rule set will be linked to by some external URL listed in the location object. What the semantics should be for rules that can not be accessed or interpreted is a matter of debate.

With regards to the specifics about rules, some requirements have been settled while others remain more fluid. One certain requirement is that rules be handled securely. Rules reveal a lot about the users who made them: their friends, personal habits, business associates, etc. For that reason alone the location server must be designed so that it only reveals rules when necessary. For the rule holder as well, authentication, encryption, and security mechanisms must be created.

The rule set format is in XML, perhaps in tabular form or some simple database structure. Each rule mentions a user or group and the granularity of location information that may be sent to them.

For simplicity, the first version of the Geopriv system only allows additive rules, for example we would create rules as in Table 1.

Because it is specifically disallowing `tom@example.com` to receive information.

With this restriction, the order in which rules are stored and retrieved will have no effect on the outcome of queries. For any particular user, one or more rules may apply and they should be combined to produce the most lenient outcome of all of the rules put together. Care must be taken that the rules are interpreted in the strictest way possible. If no rule is defined for a particular requester, the location server ought to not respond at all. Sending back an empty location object reveals presence, leaking information in a subtle way. Additionally, when designing languages for rules, it is essential that location servers ignore rules that they can not interpret. Since rules can only give permission, not remove it, skipping a rule in the evaluation will only lead to a more restrictive answer, not a more lenient one.

One proposal for XML based rule syntax has been put forward by Henning Schulzrinne [10]. His proposal includes a template for encoding civil location, a format that has been adopted by the Geopriv working group until a better solution is developed.

With the system given above, there are some issues that remain to be addressed. They are being worked on over the Geopriv mailing list which is run by the IETF. Below we discuss four of the important ones to show the places where the Geopriv system still needs to mature.

The first issue is how to provide anonymity while providing location information. When users distribute location information to their friends, family, or co-workers, they want to ensure that others can't track their movements as well. Similarly, users want to use certain sensitive location services privately, gaining from having made their location available, but not leaving themselves open to direct identification. To address these concerns, the Geopriv requirements require the possibility of using unlinked pseudonyms. An unlinked pseudonym is just a string of characters or bits that have no traceable association to the user, so a plain text field suffices for this requirement. What complicates it is the management of unlinked pseudonyms to keep them unlinkable. One entity in the system must take responsibility for keeping pseudonyms randomly generated and recycled often enough so that any particular string can't be associated with a particular user.

Ruling	Generator	Recipient	Time Restriction	Granularity	Redistribute
Allow	JohnDoe	user@example.com	Monday	State	NO
Allow	JohnDoe	*@example.com	Friday	County	YES
Forbid	JohnDoe	tom@example.com	Never	County	NO

Table 1: Sample rules for first version of Geopriv architecture

A second issue is the management of location privacy in emergencies. In the normal Geopriv system, there is delay from queries that involve going all the way back to the rule holder. Local laws require that emergency calls be answered as quickly as possible, so there must be a guarantee that the service will resolve the rule set query accurately and immediately. An additional complication comes in authentication. Normally location recipients authenticate before they receive information from the location server. However, in an emergency call, because location information is essential to the proper routing of the call, location must be provided before any sort of authentication can take place. Efficient and clear rules must be established to take care of this.

A third issue is the management and interpretation of rules. An algorithm must be designed that evaluates rule sets consistently. Care must be taken when combining fields so that rules will not combine to yield nonintuitive outcomes. For example, if we have the pair of rules in Table 2, when `joe@example.com` requests location information, simple rule combination may yield a location object that contains city information and is redistributable, certainly not the intent of the rule maker. This motivates the need for a careful, semantically correct method of combining rules. Methods for doing rule lookup and management can be borrowed from literature in the database community.

One final issue that must be addressed is security. Rule security has been addressed above in general, but care must be taken so that all protocols for communication between Geopriv entities are secure. For interoperability one secure method must be guaranteed to work for all Geopriv communications while others can be implemented as optional features.

As an aside, it is important to note the kind of privacy provided with this system: privacy by location obfuscation. Recipients may either receive no data at all when requesting location information or receive some modified data in the location object. Provision of rules to falsify location information is both an ethical and technological question that is outside of Geopriv’s scope.

5.2 Einar Snekkenes

Einar Snekkenes [11] proposes a language for writing geolocation privacy preferences as well as an architecture that supports those rules. His focus is on designing a language that can be modelled mathematically and reasoned about formally, more than one that is ready for immediate implementation. His assumptions result in a system that is less complex than the Geopriv system described above.

Snekkenes defines the following terms for use in describing his system:

- **Policy Custodian:** Entity that stores privacy policies. It may also be involved in the enforcement of policies - i.e., acting as a server that takes raw location information, applies policy, and returns a (perhaps) modified version of the location object.
- **Policy Custodian Directory:** Well known location where owners store privacy policies.
- **Location Provider:** Entity that serves out the location information.
- **Service Provider:** Entity that uses location information it gains from others to provide a service.
- **Service Consumer:** Entity that uses or consumes a service offered by the service provider.
- **Located Object:** Entity whose location information is being passed around to provide a service.

Using the above terms, Snekkenes creates the following architecture. The user has some located object, for example a cellular phone. Through communication with the wireless infrastructure, triangulation, or some other method, the network senses the location of the device and provides that data to a location provider. The user has previously defined some privacy policy that resides safely in the policy custodian directory. When a service provider queries the location provider for location information, it queries the policy custodian for the user’s privacy policy and

Ruling	Generator	Recipient	Time Restriction	Granularity	Redistribute
Allow	JohnDoe	*	None	Country	YES
Allow	JohnDoe	joe@example.com	None	City	NO

Table 2: Geopriv rule combination example

returns location information to the service provider accordingly.

Note the choice made in this architecture assuming that the network/location provider is trusted. With that it is easy to understand how the location information is created and stored. Note, however, that the privacy policy language would work just as well if this assumption were dropped.

The location data object, called an *observation* by Sneekenes, contains four pieces of information: Time of the observation, Location in some coordinate space, Speed, Identity of the located object and its owner.

These data can be viewed as a 4-tuple and manipulated as such. Each field defines some axis of location information and can be changed independently of the other fields. The rules in the privacy policies do just that.

With the above definition of an observation, the granularity and accuracy of the information in each field can be raised and lowered in a mathematically quantifiable increment. Specifically, the level of accuracy in each field can be modelled with lattice structures. By carefully defining lattices that model the accuracy of each data field, rules can be written that provably give a higher level of privacy. The advantage of this approach is its simplicity and the ease of modelling it mathematically.

Before discussing the specific lattice implementation for each field, a brief introduction to the basics of lattice structure is in order. Intuitively, a lattice may be thought of as a collection of points where some of the points are connected to other ones with straight lines. The collection has two unique elements - a bottom element and a top element. By heading down from the top element it is possible to reach every element in the set, whether directly or indirectly. By heading upward from the bottom element, it is possible to reach every element in the set, whether directly or indirectly. For each pair of points it is possible to find a greatest lower bound (GLB) and a least upper bound (LUB). Mathematically, lattices are described as a partial order over a set of points. The points may represent individual values or collections of values.

For the particular system being discussed, a lattice is defined to structure the accuracy of each field in the observation. In all the lattices, the top element is the most defined value - the exact value for the field as observed and the bottom element is the least defined value - a value or set of values that does not yield any interesting information. The lattice structure comes naturally to some of the fields, but leaves a somewhat confusing result for others. Examples of each type follow.

A lattice structure intuitively fits for the *time* field. Let t_1 be the real time of the observation and the top element of the lattice. Let the current time of the query be t_2 . The bottom of the lattice is the set of all times in the lattice less than t_2 . The lattice consists of sets of time values all less than t_2 . A partial order on points in the lattice is as follows - if the set at point A is a proper subset of the set at point B, then A is more defined than B and hence closer to the top. The intuition is that extra time points in a set are noise, meant to obscure the real value. A set that has less noise is more accurate. It is not difficult to determine from the above setup that the GLB of two points in the lattice is their union - the set that includes the noise from both sets. Similarly, the LUB for two points is their intersection - the set that has only the noise that both sets share. Using the lattice to define privacy settings will mean starting from the top of the lattice and moving down until a point with an acceptable noise level is reached.

Similar lattices can be defined for the speed and coordinate location. In general, any numeric field fits into the above lattice structure.

A lattice structure, however, is not a good model for identity. How is it possible to compare levels of identity information in a lattice? One proposal is to make a set of possible identities and hide the real identity among them, but that doesn't give an easily quantified measure of anonymity. Is hiding among 20 others more private than among 19 others? Instead Sneekenes proposes obfuscation along axes of identification - age, sex, nationality, employer, etc. - numeric or indexed fields that are more amenable to a lattice. Still, it remains unconvincing that identity is well served by the lattice model.

Using the lattices defined above for the accuracy and granularity of observations, an observation level lattice that is a combination of all the four sub-lattices can be defined. The top element is the actual observation data, the bottom element a set of anonymized data, and the GLB and LUB metrics determined by composing the GLB and LUB metrics from the sub-lattices. With the observation lattice it is possible to gain a partial ordering on all observation objects.

In addition to designing a lattice to order observations, Snekkenes suggests creating a lattice for the purposes for which service providers request location information. Purposes range from collecting anonymous statistics to services designed to track a device's every movement. Defining a good ordering on purposes is difficult and subjective, but would aid computers in making better decisions about how to handle the privacy policies.

Snekkenes defines two types of service provider requests, both of which are available in his query language.

- “Are you at location X?” This is answered in a yes/no fashion and is encoded with a one bit Boolean value.
- “Where are you located?” This requires a coordinate string for an answer, a considerably longer message.

The context of a query has the following fields:

- Current Time
- Service Provider: identity
- Service Consumer: identity
- Service Initiator: identity
- Service Requester: identity
- Purpose: What the information will be used for, indexed to a purpose ordering lattice.
- Query Type: Where are you/Are you here
- Query Expectation: Indicates that an observation object is requested.

The query context is passed to the location provider with the appropriate fields filled in. The location provider fills in the rest and passes it back.

Snekkenes' policy language can be abstracted as a function that takes an observation and an identity

as input and produces a new observation as output using the machine understandable policy as a guide. The policies are a series of pairs that have the following form: (Guard, Rule). The guard part is a filter on the context of the inquiry and can include any field data that would be in a location object - identity of the requester, current time, time of observation, geographic location of the located object, etc. More extensions are possible, but are not developed by Snekkenes.

The rule part of the pair consists of a substitution language that can replace a data field with one of the following possibilities:

- Observation from the environment (actual data)
- Some reduced accuracy version of the observation
- Greatest lower bound of two observations
- Some constructed observation (a lie)
- Observation that the requester expected (perhaps a lie)

It also is useful to have a default policy to fall back on when no other policy applies. Such a default policy could call for sending back a completely anonymous observation, something very coarse, or nothing at all.

The actual implementation of the rule language and interpreter was found to be very slow. Inefficiencies came from the need to do policy rule lookups for every query. Using a cache system to keep policies around for a reasonable amount of time on the location server before they are refreshed would solve that.

Snekkenes leaves for further work methods of optimizing the rule interpretations and making a user friendly tool to create complex and real life privacy policies. He also describes the possibility for reverse lookup - a way to ask “Who is at location X” - and to keep some kind of history to answer the question “Was I here before?” Both of these features could be implemented without changing the structure of his rule language.

5.3 Privacy Preserving Data Mining

Privacy preserving data mining [5], an approach brought by Agrawal and Srikant, is a method for discovering trends and distributions within a data set while preserving the privacy of individual fields. The process is begun with a set of private data and then

some random amount is added to each field in such a way that it is impossible to tell what the original value was. The trick is that the amounts added come from statistical distributions that allow reconstruction of the distributions from the real data just by analyzing the modified data. The application for location privacy systems is clear: Grant privacy assurance to individuals who reveal their location information by modifying their data before it is stored. In this manner the individual is safe from privacy breaches while useful trends can still be extracted from the data.

5.3.1 Related Work

Standard methods for data privatization have suffered from the same problem: the more variance introduced to retain privacy, the less useful the data becomes. This leaves customer data base owners with no choice but to either have little user privacy or work with inaccurate data. One idea to resolve this issue is query restriction where the results from a real data query are sampled, swapped around, or obscured slightly with noise. Another idea is data perturbation in which data fields are swapped between records, noise is added to each item, or fields are replaced with numbers that come from the real distribution of that field. Both methods either force managers to include incorrectness in their data or to tolerate the corruption of their data base with the swapping of data across records. In all cases private users are required to trust the database owner to keep their data private and to obscure it appropriately. Privacy preserving data mining solves these problems.

5.3.2 Methodology

The methodology recommended by Agrawal and Srikant is called value distortion. Value distortion of a field x is done by adding some value r to it. That r comes from a distribution of a particular random variable R . The value $x+r$ replaces the field x and is stored in the database in x 's place. The distribution of R can be either a uniform distribution with a mean of zero or a Gaussian distribution with a mean of zero and a standard deviation defined as desired. Agrawal and Srikant show both methods in their paper.

An alternative to value distortion is value class membership - replacing data fields with an indication of what range of values the real data fall in. Agrawal and Srikant show mathematically that of the two methods, value distortion yields the best level of

data distribution recoverability at high levels of privacy because it doesn't lose distinctions in the data the way that class membership does.

With the modified data sets, [5] explains how to recover data distribution patterns using an iterative approach and Bayes' rule. The method approximately recovers the posterior distribution function and density functions for a modified field X . The difficult part of the method is deciding when to stop. In [5], Agrawal and Srikant note that they keep calculating until successive iterations produce only very small differences in their estimates.

With this tool it is easy to design an efficient location privacy system. (For clarity the Geopriv terminology is used to describe this.) There are two ways to manage the particulars of data obfuscation:

- Have client program do it before the location data is uploaded
- Have the location server do it before the information is stored.

The decision should be made based on how much the users trust the location server.

In the first option, the location server must post the parameters for its random variable R to the location generators. Alternatively, global recommendations for parameters may be produced. When the location generator becomes aware of location information, it obfuscates the location data with an instance of the random variable and then pushes the modified data to the location server. The location server can analyze and take distributions about location information for groups of people while respecting their privacy.

In the second option, the location generator pushes the true sighting information to the location server. The location server then must obfuscate the data using its random variable. The location server then may provide the modified data to parties for analysis and distribution while protecting the privacy of its users.

Possible extensions for this system include integrating it with the above mentioned privacy systems, Geopriv and Sneekenes. In such a system users make a policy statement allowing obfuscated location information to be released to particular parties solely for the purpose of statistical survey. Location recipients would then be unable to determine the exact location of any particular user. Such a system would not be useful for users who want directed, personalized location services.

5.4 Cryptographically Enforced Conditional Access for XML

Miklau and Suciu [8] propose a system for access control of XML data using managed encryption. An XML data document, a list or a small database, is encrypted record by record with appropriate keys and then openly published. Users who have the correct secrets or passwords can then decrypt the particular portions of the data document that they have the right to see. Their technique removes the need for a secure server and secured communications, placing the access control within the document itself.

Miklau and Suciu motivate their system by arguing that certain databases are kept private, but individual records from them are revealed to particular users under certain conditions. For example, a credit card company closely guards its customers' card numbers and balances, but reveals the credit remaining on a customer's account to a merchant who queries the server with the customer's credit card number. Motivated by this example, Miklau and Suciu put forward an encryption scheme and query language that allow encrypted records to be decrypted by those with previous knowledge of the data.

5.4.1 Methodology

The atomic structures necessary for implementing their technique are as follows.

- A secure one way function, f , one that is easy to compute but very difficult to invert.
- A secure, well known encryption function E that can be inverted with well known decryption function D .

With these pieces, records are pairs of the form $(f(a), E_a(b))$. Users who know value a can search in the encrypted database for the term $f(a)$ and then decrypt $E_a(b)$. More sophisticated designs can be done, like chaining the results of one query to the keys for another query.

This technique can be extended to make conditional access rules for documents. An XML document can be viewed as a tree with a single root node and many branching child nodes. The intuition is that for a security context C , a descendant of the root, suppose users who present bindings $\{B\}$ that are descendants of C may access data at nodes $\{F\}$ that are also descendants of C . For security purposes, limit $\{F\}$ to be nodes that are descendants of C , but

not covered by any other security context. Such a system may be formulated with rules that are XPath [7] expressions. The bindings (keys) that are provided serve as the input to the rule. The expression evaluates to a set of free values that are the result of the XPath query.

With the above definition of conditional access rules and the blurring of data and inputs mentioned above, an iterative method may be designed that determines all of the nodes that a user can access based on initial inputted bindings $\{B\}$. The function begins by discovering all the unrestricted nodes and then discovers which ones become available with the initial bindings. Each iteration takes the newly discovered free values from the previous iteration, adds them to the bindings, and then explores which new free values can be discovered.

The conversion of a plain-text XML document to one encrypted in this manner is done as follows. For each conditional access rule defined, build a table of encrypted values of the type described above - $(f(key), E_{key}(value))$. For each entry, replace it with a table for each applicable rule. Replace subtrees and sub-contexts for each entry appropriately. This replacement leads to an explosion in document size proportionate to the number of rules. Miklau and Suciu propose a manner of minimizing the rules required for any particular document by viewing each rule as a function and using functional dependency elimination.

An alternative approach that uses their technique is as follows. Records could be encrypted as pairs with a password or secret being the knowledge required to decrypt records. Elaborate systems of chaining together passwords with different access rights can be built off of this technique. Those extensions are beyond the focus of the paper, but are interesting from the perspective of designing location privacy systems.

5.4.2 Security

The security of the described technique is modest because of the efficacy of dictionary and guessing attacks against the encryption.

In a guessing attack, an attacker tries random values against the database. If the key space is small enough, the attacker may be probabilistically likely to discover some valid keys after enough querying. Normally on server based systems, guessing can be tracked and blocked, but when the database is stored locally by the attacker, there is no way to control

it. The only solutions are to maximize the key space or to make the one way function slower to slow down the attacker. Note, however, that the second solution makes even legitimate accesses slower.

In a dictionary attack, the attacker tries to exhaust the key space to discover the valid keys. Such an attack may be prevented by expanding the key space and forcing good key distribution. Miklau and Suciu argue that secure server implementations also suffer from this vulnerability and therefore this shouldn't be considered a problem linked to their technique.

The above vulnerabilities argue for care when using this technique for encrypting data. It is appropriate for databases that require medium level security- encryption that can be broken with enough effort, but not easily or quickly. It would be appropriate for semi-private data like vendor inventory catalogs and data whose importance expires fairly rapidly. Highly sensitive data ought to be secured in a better fashion.

Making it into a location privacy system. Using Miklau and Suciu's technique, a location privacy system can be designed as follows. A user chooses pseudonym or secret s that can be used to identify him. The user then makes a pair $(f(s), E_s(location))$ and passes it to the location server. The location server then makes the entire database available openly. Users who wish to make their location information available give their s to a location recipient who can use it to decrypt the user's location information.

This system gives medium level security from attacks on the database itself. However, by keeping the database on a non-secure, but monitored server, a location service can provide open access to the document while watching for the patterns of a guessing or dictionary attack.

Sophisticated users would define multiple secrets for multiple levels of granularity. This can be done by giving out secret s_1 to close friends and family members and encrypting exact location information under it while giving secret s_2 to service providers and encrypt lower granularity location information under it. With this extension the notions put forward by Geopriv can be modified by defining privacy policies in terms of location granularity and then encrypting those objects to ensure proper privacy enforcement.

The issue of permission revocation remains a problem for this system.

6 Comparison of systems with respect to the scenarios

Having described in detail the four different location privacy systems that can be designed using the mentioned techniques, now this paper examine how the systems can and can not address the requirements of the three scenarios described above.

6.1 Scenario A: National Parks

6.1.1 Geopriv

The Geopriv system offers a method for reducing location information accuracy for individuals. Since the Parks system wants to track the movement patterns of its visitors rather exactly, a system that reduces the accuracy of location information while retaining identity information would not be useful. As such, personal privacy policies would not be a good fit to their requirements. An alternative to address the problem is to assign one user name for all visitors. This would give anonymity to each visitor even as the Parks track the location objects carefully. In any case, it would not work with any existing default privacy policies that have already set up and so would require a significant amount of work from the users to implement this idea properly.

6.1.2 Snekkenes

With a few tweaks, Snekkenes' privacy policy language addresses the Parks' requirements. Since his system can obfuscate personal identity down to total anonymity, it is easy for a visitor to define a policy that releases location information with all of its identity information removed. The Parks can then collect its visitor tracking information while respecting users' privacy.

6.1.3 Privacy Preserving Data Mining

A system designed with privacy preserving data mining fits the requirements of the Parks system quite well. Since the system can obfuscate location to the point where each particular datum is unintelligible, visitors retain their privacy. However, since the data are obscured in such a way that distributions can still be retrieved from them, the Parks will be able to track visitor trends accurately.

6.1.4 Conditional Access Rules

A system designed with XML conditional access controls is somewhat suited to this task. Users can define a secret that the Parks system knows and thereby allow them to access the particular location of each user without knowing the user's identity. This requires the Parks to publish a user name that all its visitors can use. Visitors who want to use their own unlinked pseudonyms/secrets to identify themselves can do that, though it leaves them open to tracking of the pseudonym.

6.1.5 Summary

Of the systems proposed, only the privacy preserving data mining approach can be used without modification. Both the Snekkenes and the conditional access control systems require users to make some modifications to their privacy policies, a step that may stop a significant amount of data collection from lazy or uninterested users. A system that guarantees personal privacy but doesn't require specific user modification of existing policy is the ideal.

6.2 Scenario B: Rental Car Company

6.2.1 Geopriv

The Geopriv system addresses the needs of the car rental agency as follows. First assume that the renters have previously defined a privacy policy about how car rental agencies may access their data. The user can design a privacy policy along the following lines to answer each of the three types of data wanted:

- The policy in Table 3(a) returns a very low granularity information when the car is not in one of the rental agency's lots. The agency is concerned about inventory control only when its cars are returned, not when they are out being used by customers.
- The policy in Table 3(b) reveals state level information when the car is being moved by the customer. That will allow the rental agency to enforce its policy of the car not crossing state lines without violating the customer's privacy.
- A policy to reveal exact usage patterns of the car is hard to describe in generic policy terms. The policy in Table 3(c) is a first cut is to allow information about the name of the road that the user is on, but that could be a breach of privacy

in some situations. Ideally what is needed is a way to describe *local street* as opposed to *highway* but given Geopriv's current draft for civil location information that is not possible.

Geopriv provides a concrete language that allows users to make policies to take care of two of the rental company's requirements. The first one requires a way of determining when the car is parked in a car lot, not a simple task. The second type fits very nicely into the Geopriv model. The third could be expressed in Geopriv, but requires a change in the language for civil location description.

6.2.2 Snekkenes

Snekkenes' language addresses the three types of information as follows:

- For tracking car inventory, the language allows for rules based on proximity to other located objects. Users can therefore make a policy that allows the car's location to be known when the car is within a reasonable distance of the car rental lot and not in motion. Except for some unusual circumstances that will maintain the user's privacy. For the odd case where the driver is stuck in traffic near a car rental lot, the policy can add the requirement that the driver's cell phone not be in the car, a reasonable indicator about whether the user is in the car.
- For tracking the state that the car is in, a policy can be written allowing the car rental company access to state level information at any time.
- For discerning car usage patterns, users can define a location accuracy matrix that includes one level of road type. Since Snekkenes doesn't give any particulars about how he designs his lattices, this could be made a requirement.

Snekkenes' policy language can be used to define requirements for all three types of information that the rental agency wants. His language has the advantage of allowing conditions in terms of proximity to landmarks as well as other located objects. That gives his language the flexibility that is needed for this scenario.

6.2.3 Privacy Preserving Data Mining

A location privacy system based on privacy preserving data mining addresses the three types of information as follows:

Policy (a):

Ruling	Generator	Recipient	Location Restriction	Granularity	Redistribute
Allow	JohnDoe	Rental Agency	Not in car lot	Country	NO
Allow	JohnDoe	Rental Agency	In car lot	Exact	YES

Policy (b):

Ruling	Generator	Recipient	Location Restriction	Granularity	Redistribute
Allow	JohnDoe	Rental Agency	None	State	NO

Policy (c):

Ruling	Generator	Recipient	Location Restriction	Granularity	Redistribute
Allow	JohnDoe	Rental Agency	In motion	Road Type	NO

Table 3: Rental Car Geopriv policies

- For tracking car inventory location there isn't much hope. The methods of privacy preserving data mining won't give accurate enough information to be useful for tracking. The only solution is for the user to turn off tracking of the car for the term of the rental.
- For tracking the state that the car is in, finer grained location information can be obfuscated and only state level information be released. The rental company doesn't care about the finer grained location of the driver so this method will suffice.
- For tracking the usage patterns of the car, the location of the car can be described just in terms of "street" or "highway," obfuscating the rest of the information.

A privacy preserving data mining system doesn't have much to offer to these requirements. The solutions described bypass the features of privacy preserving data mining and require leaving essential information in clear text. Specific information is required since the rental company wants exact information about cars.

6.2.4 Conditional Access Rules

A system based on conditional access rules addresses the three types of information as follows:

- For tracking car inventory, to ensure privacy during the rental, the user encodes the car's location information with some secret that the car rental company doesn't know. The user removes the secret upon termination of the rental.
- For tracking the state that the car is in, the user encrypts the car's location at the state level using a secret that the rental company knows. This

way only the company can decrypt the state information.

- For tracking the usage patterns of the car, by defining an encoding to describe "street" and "highway," the user can encode that information with some secret that the rental company knows and then release it. This way only the rental company can track the car's movement. In order to keep the car information specific, the user must either encode identity information in the data or use a unique secret that the rental company can track.

The conditional access rule system is the means of securing the data that is being provided. The system doesn't give a method for how to describe the desired data. The advantage of this system is primarily its encryption and security that allow only the rental agency to see the data.

6.2.5 Summary

Of the systems proposed, both Sneekenes' system and a conditional access rule based system fulfill the requirements. Both give good control over limitations on access and data accuracy. Geopriv can fulfill most of the requirements, but is hampered by its currently policy for civil location information. Privacy preserving data mining has an entirely different scope and so has little to no usefulness for this scenario.

6.3 Scenario C: Location Services

6.3.1 Geopriv

The Geopriv system is well suited to making a location information system for this scenario. In particular, travellers who want information about ser-

vices and stores near them can define a privacy policy that allows certain chain stores to follow their locations and others not to. In order to receive the most accurate service information, they must reveal accurate location information. Users make privacy policies defining the redistribution rights of the location object, but there is nothing to control the kind of data that is sent to the user based on the location information provided. Managing sensitive information like appointments and directions requires a secure method of communication, something that has not been well addressed in the Geopriv documentation yet.

The goal of Geopriv has been one way communication of location information. Its language is a method for a rule maker to describe how location information is passed to recipients. It doesn't say anything about what the data may be used for because the system is still in its infancy. Defining usage rules for location information would be an interesting extension of the Geopriv rule set.

6.3.2 Snekkenes

Snekkenes' language for location privacy management yields results similar to the Geopriv system. It is easy to write a privacy policy that describes how traveller location information is accessed and used. Since Snekkenes has an architecture that discovers location by a trusted network, the network itself can hold the repository for sensitive personal information that is to be served to the user. This eliminates problems with the transfer of information across insecure channels. Users make privacy policies declaring how and when they can be contacted and what level of accuracy of location can be delivered. Just as with Geopriv, it remains an open issue how to assure that outside service providers use the data only for authorized messaging, not spam.

6.3.3 Privacy Preserving Data Mining

Because the concern is not with extracting trends in data, but rather in delivering personalized location information, privacy preserving data mining doesn't fit these requirements.

6.3.4 Conditional Access Rules

Conditional access rules are useful for these requirements. By publishing a document or making available a whole set of encrypted location data, users

can be sure that they are only tracked by those who have the right secret and are authorized to see the data. Users create and exchange secrets or pseudonyms that allow certain entities to track their locations while keeping their location private from others. Users may release several versions of their location data with several different granulations, making different levels available to parties as appropriate. Private information about meetings and maps can be passed to the user encoded with the shared secret, simplifying the communication between the user and the location information system. The full features of the conditional encryption system can be used to release data according to privilege level and the preferences of the user.

6.3.5 Summary

In summary, this scenario brings out the best features of the Geopriv and Snekkenes systems. It is along the lines of both papers' architectural goals and so is well addressed by them. Since privacy preserving data mining is solving a different problem, it doesn't do well here. Conditional access rules serve well for defining rights and encryption on data, allowing for a rich set of options for releasing location data.

7 Future Work

One common theme that emerges from all of the scenarios is that the onus for designing policy is on the user's back. Almost all situations require that the user carefully construct a policy that is appropriate for the task at hand. As a language gets complicated, the number of users that can properly configure and maintain a dynamic rule set gets smaller. In this lies one of the hardest problems facing location systems design. Ideally users should make a set of rules once and let that set persist and suffice for the longer term. Doing that requires a way of making generic statements about different recipients, taking a step back from making permissions based on individual users and entities. Developing such a generic policy language is real world challenge to all of the systems.

Another place where work is needed is making tools that allow users to easily create and edit policies. It is not realistic to expect users to hand-code XML rule sets or conditional access XPath expressions. It is up to systems developers to create intuitive tools to make policy creation easy for end users.

Finally, both Geopriv and Snekkenes include in

their architectures a central location to store and retrieve user privacy policies. Neither makes specifications about the design, security, or ownership of such a location. It is important to decide how to organize a policy storage site to be always available, immune to denial of service attack, able to compartmentalize user data, and to secure policies from unauthorized access. Whether the storage facility is owned by individuals, groups, ISPs, or companies, is a problem that also must be solved before any viable location privacy system can be completed.

References

- [1] Internet Engineering Task Force. www.ietf.org.
- [2] Geographic location/privacy (geopriv). 2003. www.ietf.org/html.charters/geopriv-charter.html.
- [3] The original bluejacking web site - what it is, how to do it, tips, tricks, stories, photos. 2003. www.bluejackq.com.
- [4] Webopedia. 2003. www.webopedia.com.
- [5] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 439–450. ACM Press, 2000.
- [6] Jorge Cuellar, Jr. John B. Morris, Deirde Mulligan, Jon Peterson, and James Polk. Geopriv requirements - draft-ietf-geopriv-reqs-04.txt. Work in progress, 2003.
- [7] Georg Gottlob, Christoph Koch, and Reinhard Pichler. XPath processing in a nutshell. *SIGMOD Rec.*, 32(2):21–27, 2003.
- [8] G. Miklau and D. Suciu. Cryptographically enforced conditional access for xml. In *Proceedings of WebDB*, 2002.
- [9] BBC News. New mobile message craze spreads. 2003. news.bbc.co.uk/1/hi/technology/3237755.stm.
- [10] H. Schulzrinne, J. Morris, H. Tschofenig, J. Cuellar, and J. Polk. Policy rules for disclosure and modification of geographic information - draft-ietf-geopriv-policy-00.txt. Work in progress, 2003.
- [11] Einar Sneekenes. Concepts for personal location privacy policies. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 48–57. ACM Press, 2001.
- [12] W3C. Extensible markup language (XML). 2003. www.w3c.org/XML/.
- [13] David Whaelan. The unofficial cookie FAQ. 2002. www.cookiecentral.com/faq/.